IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Antonio Asaro et al.	Examiner: Paul R. Myers
Application No.: 10/074,064	Group Art Unit: 2111
Filed: February 12, 2002	Docket No.: 00100.00.0130
For: **METHOD AND APPARATUS FOR A DATA BRIDGE IN A COMPUTER SYSTEM**	

APPEAL BRIEF PURSUANT TO 37 C.F.R. § 41.37

Dear Sir:

Appellant submits this brief further to the Pre-Appeal Brief Request for Review filed May 21, 2008, and the Notice of Panel Decision from Pre-Appeal Brief Review dated June 18, 2008 in the above-identified application.

TABLE OF CONTENTS

Page

I. REAL PARTY IN INTEREST 3

II. RELATED APPEALS AND INTERFERENCES 4

III. STATUS OF CLAIMS 5

IV. STATUS OF AMENDMENTS 6

V. SUMMARY OF CLAIMED SUBJECT MATTER 7

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 16

VII. ARGUMENT 17

THE 35 U.S.C. § 103(a) OBVIOUSNESS REJECTIONS MUST BE REVERSED SINCE THE REFERENCES DO NOT TEACH WHAT IS ALLEGED 17

THE 35 U.S.C. § 103(a) OBVIOUSNESS REJECTIONS MUST BE REVERSED SINCE THE EXAMINER DID NOT PROVIDE A FACTUAL BASIS FOR THE GRAHAM FACTORS 24

VIII. CONCLUSION 24

CLAIMS APPENDIX/CLAIMS ON APPEAL ....APPENDIX A

EVIDENCE APPENDIX ....APPENDIX B

RELATED PROCEEDINGS ....APPENDIX C

I. REAL PARTY IN INTEREST

ATI International SRL is the real party in interest in this appeal by virtue of an executed Assignment from the named Inventors of their entire interest to ATI International SRL. The Assignment evincing such ownership interest was recorded on February 12, 2002, in the United States Patent and Trademark Office at Reel 012593, Frame 0977.

II. RELATED APPEALS AND INTERFERENCES

To Appellant's knowledge, there are no related Appeals or Interferences filed, pending, or decided.

III. STATUS OF CLAIMS

Claims 1-28 and 33-35 are pending. Claims 36, 30 and 31 have been allowed. Claims 1-28 and 33-35 stand rejected. The originally filed Application contained claims 1-31. Claims 32-36 were added during prosecution. Claims 29 and 32 were canceled during prosecution . Claims 1, 2, 10, 11, 28, 33, 34 and 35 were amended during prosecution of the present application. Of the pending appealed claims, 1, 10, 19, and 28 are independent. For purposes of this appeal only, claims 2-9 stand or fall with claim 1; claim 34 does not stand or fall with claim 1 but is patentable on its own. Claims 11-18 and 33 stand or fall with claim 10. Claim 35 does not stand or fall with claim 10 but is patentable on its own. Claims 20-27 stand or fall together with claim 19.

IV. STATUS OF AMENDMENTS

A Pre-Appeal Brief Request for Review was filed on May 21, 2008, in response to the Final Office Action mailed on November 21, 2007. Amendments under section 37 CFR § 41.33(a) were made to the claims, subsequent to the Final Office Action to rewrite allowable claims 29-31 in proper form for allowance. Claims 36, 30 and 31 have since been allowed by advisory action dated October 29, 2008. The claims listed in Appendix A reflect the remaining rejected claims as they stood at the time the Pre-Appeal Brief Request for Review was filed.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In one example, a ROM in a data bridge contains initial values that can be stored as initial content of a base address register (BAR) and the ROM also contains a mask value that can be used to form or configure a BAR register to be read and/or writeable. (Published Specification, at least para. 22-27, FIG. 3). The configurable register includes register configuration logic and at least one register flop to contain the initial value if desired. The at least one mask value is used to generate a mask bit for the configuration logic. *Id.* The register configuration logic configures the at least one register flop to be read and/or writable based on the mask value stored in the memory (e.g., ROM). *Id.*

As background, FIG. 1 (below) is a prior art computer system that includes a central processing unit (CPU) 1000 operably coupled to local cache 1002 and to a north bridge 1004. (Specification paragraph 2) The central processing unit 1000 when executing a memory transaction (e.g., a read from memory command, a write to memory command, or a read/write command) internally processes addresses associated with the transaction in virtual, or linear, address space. *Id.* To communicate with the north bridge, the central processing unit converts the virtual addresses into physical addresses. The north bridge, upon receiving the physical addresses, determines whether the transaction is addressing a location within a DRAM address space, or a PCI address space.*Id.*

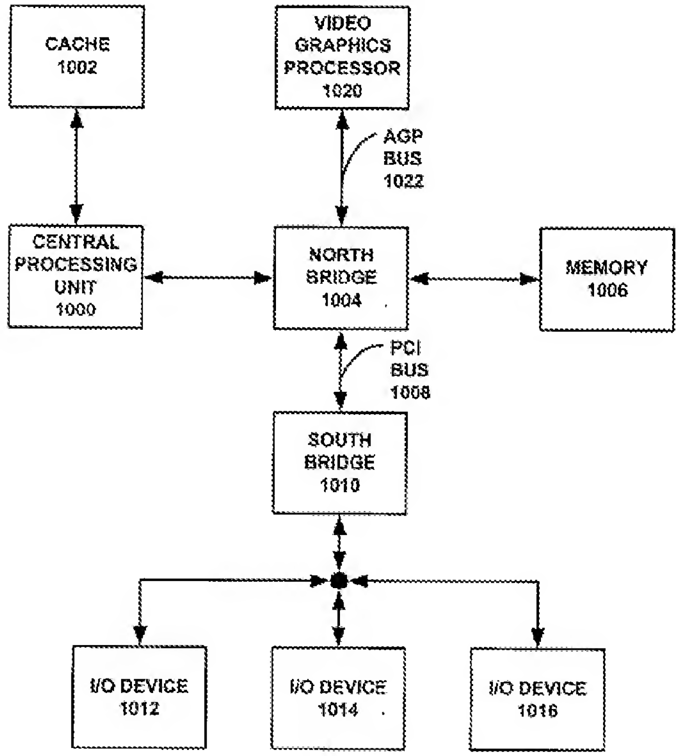

FIG. 1
PRIOR ART

If the received physical address corresponds to the GART address space, the north bridge 1004 may further translate the physical address, using a GART table, into a corresponding physical address. *(specification, para. 3)* Having obtained the physical address, the north bridge 1004 communicates with a memory 1006 to retrieve an appropriate memory block (e.g., line of memory, or multiple lines of memory where a line is 32 bits, 64 bits, 128 bits, etc.). *Id.* If the physical address corresponds to the memory 1006, the north bridge 1004 utilizes the physical address to facilitate the memory transaction. As such, if the memory transaction was a read transaction, the north bridge 1004 facilitates the retrieval of the corresponding memory line or lines from memory 1006 and provides them to the central processing unit 1000. If the received

physical address corresponds with the PCI address space, the north bridge 1004 passes the transaction to the PCI bus 1008.

Id. The south bridge 1010, upon receiving a physical address, determines which of the plurality of I/O devices 1012, 1014, 1016 is to receive the transaction. *(para. 4)*

In addition to the north bridge 1004 receiving transactions from the central processing unit 1000, it may also receive transactions from a video graphics processor 1020 and the south bridge 1010 relaying transactions from I/O devices 1012, 1014, 1016. *(para. 5)* Such transactions have varying requirements. For example, transactions from the central processing unit 1000 and video graphics processor 1020 are typically high speed transactions which require low latency. The amount of data in such transactions may vary but is generally a memory line or plurality of memory lines per transaction. The transactions from the I/O devices 1012, 1014, 1016 are generally large amounts of data (i.e., significantly more than several memory lines of data), but are typically latency tolerant. *Id.*

The video graphics processor 1020 provides display data to a display (not shown). *(para. 6)* Typically, the video graphics processor 1020 will include a frame buffer for storing at least part of a screen's worth of data. *Id.*To minimize the size of the frame buffer or to extend the memory used for generating the display data, the video graphics processor 1020 often uses system memory. Sometimes these processes are inside the AGP aperture. In this instance, the video graphics processor 1020 is writing to and reading from the memory 1006 via the AGP 1022 bus and the north bridge 1004. The processing of video graphics data requires a high speed

low-latency transmission path. *Id.*

As is known in the prior art, upon initialization of the system, each application specific integrated circuit (ASIC), such as the graphics processor 1020, has base address registers which define the starting address of a program, table, memory structure, set of registers etc. *(para. 7)* An ASIC is a chip that is custom designed for a specific application rather than a general purpose chip, such as a microprocessor. The use of ASICs improves performance over general purpose CPUs, because ASICs are hard wired to do a specific task and do not include the overhead of fetching and interpreting stored instructions. There are many varieties of ASICs. *Id.*

In order to determine the system configuration upon initialization, the computer system reads and queries the system components to determine which resources are required. *(para. 8)* For example, some components have address ranges that they would like to have mapped into system's address spaces. In PCI systems, this information is communicated via base address registers (BARs). In the prior, art this has been accomplished by the use of a limited number of straps or in PC's is most commonly configured using BIOS routines and specialized hardware in each ASIC to support these routines. *Id.*

In the prior art, information configuration data for initialization was provided by what are referred to as straps *(para. 18)*. The straps have voltage levels which set initial values for the configuration bits that are used at initialization of the system. *Id.*These configuration bits tell the system the identity of components and how they are to be setup in the system. Thus, an ASIC can come up in different modes upon initialization or resetting of the computer system. For example, if two straps are used, then there are four possible configuration settings. Such settings may be, for example, memory requirements of one megabyte, 2 megabyte, 4 megabyte or 8

megabytes. *Id.* It is obvious that the number of different configurations is limited by the number of straps that are provided. In modern day computer systems, this becomes a severe limitation with the variety of different ASICs that may be utilized with a computer system. *Id.*

FIG. 3 (below) shows one example of a circuit for providing a configurable register using a single bit of a single register, such as a base address register, a command register, configuration register or any other suitable register (i.e., memory element as disclosed by Applicants). *(para. 24)* The circuit includes a mux 360, AND gate 370, NAND gate 380, OR gate 390, Register flop 400 and Mask flop 410. The register bit is completely flexible, in that the register bit can be configured as either a read only bit, or read/writeable bit, and can be loaded with any initial value. *Id.* If the bit is read only, then this initial value will be the value that is always read from the bit. So the function of the bit (read/RW) and read only value are totally programmable, even though fixed hardware is used to form these registers. The data for the initial values and mask values is stored in something that is modifiable, for example, the ROM 215. *Id.*

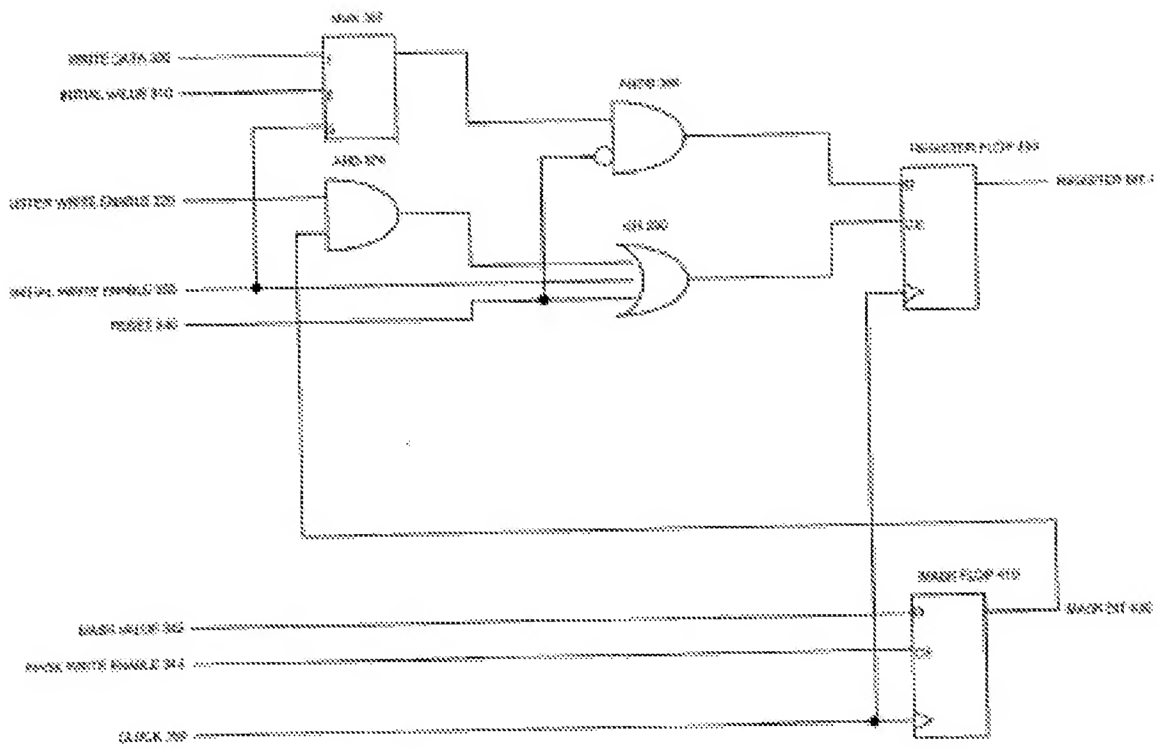

FIG. 3

The initial values and mask values are for use in forming a register. *(para. 25)* The circuit is a configurable register that includes register configuration logic (e.g., mux 360, AND gate 370, NAND gate 380 and OR gate 390). *Id.* The register flop 400 contains an initial value that is represented by the register bit 420. The mask flop 410 generates a mask bit for the configuration

logic. The register configuration logic configures the register flop 400 to be read and or writable based on the mask value 342 stored in the memory. *Id.*

Reset 340 is used to put the register bit into a defined state. *(para. 26)* The Mask register flop 410 is not reset, because, as long as the Mask register bit 430 is written to before the Register flop 400 is written by the system write path 320, the initial undefined state of the Mask register flop 410 will not propagate to the Register flop 400. Certainly the Mask register flop could be reset as well. *Id.*

At some time before the register value is read, the initial and mask values are written. *(para. 27)* The order is immaterial. The Mask register holds the mask value. When the Mask bit 430 is set to `1`, the register bit will be writeable via the system write path 320, when it is `0`, the register bit is not writeable, and the register bit is read only. *Id.*

The initial value and mask values 218 are stored in the ROM 215. *(para. 28)* These are brought in from the ROM 215 via the ROM controller 216 which generates the protocol to read data from the ROM 215. . *Id.* The ROM can be read, and the values stored in some intermediate location, if that is needed. *Id.* The initial and mask values 218 can also be read from the ROM 215 at the time that these are written to the Mask 410 and Register 400 flops. *Id.* More precisely, the mask value is read from the ROM 215 via the ROM controller 216 and is written to the Mask register flop 410 via the Mask Write Enable signal 344. The initial value is also read from the ROM 215 and the value for this bit is written into the Register flop 400 via the Initial Value signal 310 using the Initial Write Enable signal 330. *Id.*

Independent claim 1 is directed to a data bridge system *(FIG. 2)* having an interface for transferring data, a plurality of application-specific integrated circuits (ASICs) (208, 1012-1016),

and a data bridge (214) operatively coupled to each: of the interface and the plurality of ASICs. The data bridge has a read only memory (215) for storing at least initial values and mask values (218) for each ASIC of the plurality of ASICs. *(FIG. 2, para. 22, 23)* Dependent claim 34 requires that the data bridge system comprises at least one configurable register (FIG. 3) that includes register configuration logic (360, 370, 380 and 390) and at least one register flop 400 to contain an initial value and at least one mask flop 410 that generates a mask bit 430 for the configuration logic. The register configuration logic (360, 370, 380 and 390) configures the at least one register flop 400 to be read and/or writable based on at least one mask value 342 stored in the memory.

Independent claim 10 is directed to a method for configuring a plurality of ASICs that comprises reading, initial values and mask values 218 from the read only memory 215 *(para. 22, 23)*; and forming, from the initial values and the mask values, configurable registers *(FIG. 3)*, that upon initialization of a system cause the system to allocate resources to each of the plurality of ASICs. *(208, 1012-1016)* Dependent claim 35 requires that forming the configurable registers comprises forming at least one register flop 400 of the configurable register *(FIG. 3)* to be read and/or writable based on at least one mask value 410. *(para. 24-28)*

Independent claim 19 is similar to claim 1 but somewhat broader in scope and is directed to a data bridge system that includes an interface for transferring data, a plurality of components *(208, 1012-1016)* and a data bridge 214 operatively coupled to each of the interface and the plurality of components wherein the data bridge includes a storage device for 215 storing at least initial values 218 and mask values for each component of the plurality of components.

Independent claim 28 is directed to a circuit that includes memory (e.g., ROM) containing initial values and mask values for use in forming a register; and at least one configurable register (FIG. 3) that includes register configuration logic and at least one register flop to contain an initial value and at least one mask flop that generates a mask bit for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based on at least one mask value stored in the memory.

The cited references to not teach the data bridge with initial values and mask values as claimed nor the configurable registers and corresponding control logic as claimed.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 4-9, 19, 22-23 and 25-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,859,987 (Gillespie) in view of U.S. Patent No. 6,094,699 (Surugucchi) and U.S. Patent No. 5,857,083 (Venkat). Claims 2-3, 20-21, 24 and 33 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,859,987 (Gillespie) in view of U.S. Patent No. 6,094,699 (Surugucchi) and U.S. Patent No. 5,857,083 (Venkat) and further in view of Applicants admitted prior art. Claim 28 stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,675,292 (Prabhu) in view of what is well known in the art. Claims 10-11, 13, 15-17 and 34-35 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,859,987 (Gillespie) in view of U.S. Patent No. 6,094,699 (Surugucchi) and U.S. Patent No. 5,857,083 (Venkat) as applied to claim 1 and further in view of U.S. patent No. 6,675,292 (Prabhu). Claims 14 and 18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,859,987 (Gillespie) in view of U.S. Patent No. 6,094,699 (Surugucchi), U.S. Patent No. 5,857,083 (Venkat) and U.S. patent No. 6,675,292 (Prabhu) as applied to claim 10 and further in view of Applicants' admitted prior art.

VII. ARGUMENT

The references and claim language cannot be mischaracterized in an effort to render a claim unpatentable (See e.g., *In Re Rouffet*, 149 F. 3d. 1350, 47 USPQ 2d 1453 (Fed. Circ. 1998, *In Re Fine*, 837 F.2d 1071)). If a reference does not teach what is alleged, a prima facie case has not been made. (See MPEP sections 2142-2144 and cited cases). Also, there must be fact findings relating to the *Graham v John Deere Co.*, 383 U.S. 1 (1966) factors (See also, *KSR International Co. v. Teleflex Inc. et al.,* 127 S. Ct. 1727 (2007), *rev'd and remanded*, Teleflex Inc. v. KSR Int'l Co., 2007 WL 2045626, at *1 (Fed. Cir. 2007)). For one or more of these reasons the rejections should be reversed and the claims should be allowed.

THE 35 U.S.C. § 103(a) OBVIOUSNESS REJECTIONS MUST BE REVERSED SINCE THE REFERENCES DO NOT TEACH WHAT IS ALLEGED

Taking independent claim 28 first, this claim stands rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,675,292 (Prabhu) in view of well know prior art. The reference does not teach what is alleged in the office actions and as such the Examiner's position must be reversed. Claim 28 requires a circuit comprising,

> memory containing initial values and mask values for use in forming a register; and at least one configurable register that includes register configuration logic and at least one register flop to contain an initial value and at least one mask flop that generates a mask bit for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based on at least one mask value stored in the memory.

Applicants respectfully submit that it appears that the Prabhu reference has been misapprehended. Prabhu is directed to an exception handling operation for SIMD floating point instruction using a floating point status register to report exceptions – not to data bridge systems and not to the claimed subject matter. In particular, column 5, lines 40-56 of Prabhu has been cited as teaching a circuit having a configurable register that includes register configuration logic

and at least one register flop that contains an initial value and at least one mask flop that generates a mask bit for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based upon the at least one mask value. (See page 7 of final rejection). However, the cited paragraph actually refers to a *preconfigured* register block that as known in the art may have different types of registers namely general purpose registers, status registers and control registers. Some of the general purpose registers are only visible to an integer execution unit and some are only visible to graphics unit. In addition, the status registers and control registers, as known in the art, contain condition and control codes relating to the processor's operation. The last sentence of the paragraph states "although some status and control registers can be modified by program instructions, [so that they can be written to or read from] many registers may be configured as read only." (Column 5, lines 54-56). The use of the word "configured" in this context refers to the fact that status and control registers may be preconfigured at manufacture to be both read and write registers but some may be read only registers. There is no register configuration logic described because there is no configurable register. The office action alleges that the claimed configuration logic is apparently inherent. Applicants respectfully submit that it is not inherent since the reference does not teach any logic in a circuit that configures configurable registers to be read and/or write registers that utilizes initial volume and mask values as claimed. Prabhu describes *a preconfigured register block*. Even if the register block is not consider or preconfigured, *there are no mask flops that generate a mask bit for any configuration logic in Prabhu as alleged* and there is no part of Prabhu that has been cited for teaching such structure.

The claim requires that the configuration logic configures the register flop to be read and/or writable based on at least one mask value stored in the memory. The cited portion of

Prabhu again does not provide for any configuration logic that configures the register flop to be writable as opposed to readable based on at least one mask value stored in the memory but instead refers to standard circuits that simply have register blocks wherein some of those registers are fabricated as read only registers and others are fabricated to be read and writable when they are manufactured. Prabhu uses preformed registers some of which may be read only even though they are status or control registers as described by Prabhu.

In addition, the Advisory Action sums up the Examiner's position by stating that Applicants' arguments are not persuasive since "populating with data is configuring." (Advisory Action, page 2). However, Applicants respectfully submit that there is a difference between populating a register with initial values and configuring a register to make it a read or writeable register as claimed. Applicants respectfully submit that the claims themselves require something very different than simply populating a register with data. For example, claim 28 requires memory that contains both initial values and mask values for use in forming a register and in addition, a configurable register that includes register configuration logic and at least one register flop to contain an initial value and at least one mask flop that generates a mask that for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based on the at least one mask value stored in the memory. As such, the configuration described in the claim refers to configuring a register to be read and/or writable – not populating the register with a value. Since the reference does not teach what is alleged and does not teach the claimed subject matter, the Examiner's position must be reversed.

Claims 1, 4-9, 19, 22-23 and 25-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Gillespie in view of Surugucchi and Venkat. As to claims 1 and 19, the final

Office Action admits that the Gillespie reference does not teach that a data bridge read only memory stores at least initial values and mask values for each ASIC of a plurality of ASICs. Applicants also respectfully submit that the Office Action appears to mischaracterize Gillespie for failing to teach other subject matter. For example, the Office Action alleges that as to Gillespie, the base address registers "would inherently need to be configured." Applicants respectfully submit that what Gillespie would actually inherently need is that base address registers be populated with data values. Gillespie would not need to have any base address registers "configured" as claimed, for example, to be configured to be read or writable based on read/write mask values that are stored in read only memory. In their response to final action, Applicants respectfully requested a showing as to where such inherency is taught or why the configuration as to readability or writability of the base address registers would be required in Gillespie. As known in the art, the base address registers merely contain data identifying the base address register. There is no need for mask values that configure registers to be read and or writable. Applicants request for a showing was not addressed.

Also the Applicants respectfully submitted that the Surugucchi reference appears to have been misapprehended. Surugucchi is directed to a different system from that taught by Gillespie or Applicants' claimed invention. In fact, there is no read only memory used in Surugucchi and Surugucchi utilizes separate bridge circuitry and separate BASS circuit in order to simply populate configuration registers. The use of the word "mask register" as specifically described in Surugucchi is a register, not RAM, and also merely stores contents which are values—not mask values—that set the memory base address register to contain initial address location information. The mask values in Surugucchi have nothing to do with creating a register to be writable or readable.

As stated in Surugucchi, the bridge assist device (BASS) is an exposed PSI device that is used to acquire the system memory address space required by interface controllers. The memory address space is then partitioned by a controller and assigned to each of the hidden PCI devices. The BASS appears to the host system as a device with a memory space requirement which is the sum total of the space needed by the interface controllers. In realty, the BASS is a dummy device and does not use disk space. (Column 2, lines 52-60).

The BASS uses two configuration register spaces 258 and 260. As specifically stated in column 10 of Surugucchi in reference to the word "memory mask register," the contents of this register are used to set the memory base address register contents which are initial values and not mask values that change a base address register to be read or writable. As specifically stated in column 10:

> "The first configuration register space 258. This memory space is used by the interface controllers 218 to communicate with the local CPU 214. The base address of the memory space is set in the PCI memory base address register (offset 10h) and the first configuration register space 248 (see Fig. 10). The BASS control logic unit 254 then updates the corresponding register (i.e., PSI memory base address register, offset 10h). In the second configuration space 260 with the same value . . . (column 10, lines 5-13).

As such, Surugucchi uses the word mask to refer to actual initial values or contents of a register as opposed whether the register is a read or writable register. As such, Applicants respectfully submit that the Surugucchi reference teaches a known prior art systems that 1 – does not include a ROM in any data bridge and 2 – merely use the contents of one register to be written as the contents of a base address register.

In addition, the Office Action alleges that the corresponding bridge in Surugucchi is 210 and 212 alone or 212 taken together with 210. Again, Applicants respectfully note that the

bridge 210 is not a bridge to a whole system but is a bridge to a local CPU and that the configuration register space of the local PCI bridge contents thereof, such as the base address register, is populated by the BASS 212. Again, which is not ROM and which only includes initial values for the registers in the local to PCI bridge.

As such, since the configuration space in Surugucchi is used to "write data into the first configuration register space and to read from the first configuration register space" (column 3, lines 11-16) and since Surugucchi describes the "mask register" as actually containing data which serves as the contents of the base address register, and hence initial values of the base address register of the local bridge, substituting this teaching in Gillespie would merely result in the ROM of Gillespie storing initial base address register values.

As to claims 1 and 19, as used in the claims, Applicants distinguish between "initial values" and "mask values" in the read only memory. They are different information used for different purposes and control the registers in a different manner since the initial values are the contents stored in the registers and the mask values configure the registers to be, for example, read and/or writable. No such teaching or disclosure is set forth in either Gillespie or Surugucchi and in fact Surugucchi in terms of base address register population, teaches no more than what Gillespie teaches and Gillespie admittedly does not teach the claimed subject matter. Since the references do not teach what is alleged and do not teach the claimed subject matter, Applicants respectfully submit that the Examiner's finding must be reversed.

Claims 10-11, 13, 15-17 and 34-35 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Gillespie in view of Surugucchi and Venkat as applied to claim 1 and further in view of Prabhu. As to independent method claim 10, the claim requires reading, initial values

and mask values from a read only memory; and forming, from the initial values and the mask values, configurable registers, that upon initialization of a system cause the system to allocate resources to each of the plurality of ASICs. (emphasis added). This claim is also rejected based on the same grounds as claim 28 with respect to Prabhu. Prabhu again (nor do any of the cited references as evidenced by the allowed claims) does not teach forming configuragle registers from initial values and mask values as claimed but instead refers to standard circuits that simply have register blocks wherein some of those registers are fabricated as read only registers and others are fabricated to be read and writable when they are manufactured. Prabhu uses preformed registers some of which may be read only even though they are status or control registers as described by Prabhu. The Examiner's position that "populating with data is configuring" or forming (Advisory Action, page 2) is contrary to the references and to the claim language. Applicants respectfully submit that there is a difference between populating a register with initial values and forming a configurable register to make it a read or writeable register as claimed. The Examiner's rejection must be reversed.

Claim 34 further requires the data bridge system of claim 1 wherein the configurable register includes register configuration logic and at least one register flop to contain an initial value in at least one mask flop that generates a mask bit for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based on at least one mask value stored in the memory. Claims 34 and 35, are rejected on the same grounds as claim 28 in view of Prabhu and accordingly Applicants respectfully resubmit the remarks made above with respect to Prabhu and claim 28. Therefore the Examiner's rejection must be reversed as to claims 34 and 35.

THE 35 U.S.C. § 103(a) OBVIOUSNESS REJECTIONS MUST BE REVERSED SINCE THE EXAMINER DID NOT PROVIDE A FACTUAL BASIS FOR THE GRAHAM FACTORS

The Examiner also failed to provide a prima facia case of obviousness since the Examiner has not identified the level of ordinary skill in the art that the Examiner is using in an attempt to render the claims unpatentable. Since there is no fact finding related to at least this fact required under the Graham v. John Deere Company, 383 US1(1966) Appellant respectfully submits that the rejections must be withdrawn. (See also, *KSR International Co. v. Teleflex Inc. et al.,* 127 S. Ct. 1727 (2007), *rev'd and remanded*, Teleflex Inc. v. KSR Int'l Co., 2007 WL 2045626, at *1 (Fed. Cir. 2007)).

VIII. CONCLUSION

For the reasons advanced above, Appellant submits that the Examiner erred in rejecting pending claims 1-28 and 33-35 and respectfully requests reversal of the decision of the Examiner.

Respectfully submitted,

Date: November 21, 2008

By: /Christopher J. Reckamp/
Christopher J. Reckamp
Registration No. 34,414

Vedder Price P.C.
222 N. LaSalle Street
Chicago, Illinois 60601
PHONE: (312) 609-7599
FAX: (312) 609-5005

CLAIMS APPENDIX

CLAIMS ON APPEAL

1. A data bridge system, comprising:

an interface for transferring data;

a plurality of application-specific integrated circuits (ASICs);

a data bridge operatively coupled to each: of the interface and the plurality of ASICs; and

the data bridge having a read only memory for storing at least initial values and mask values for each ASIC of the plurality of ASICs.

2. The data bridge system according to claim 1, wherein at least one of the plurality of ASICs is a graphic processor.

3. The data bridge system according to claim 1, wherein the interface is connected to a north bridge in a computer system.

4. The data bridge system according to claim 1, wherein the data bridge upon initialization forms at least one of: base address registers that are queried by the interface, command registers, and configuration registers.

5. The data bridge system according to claim 4, wherein the data bridge has six base address registers, each of the base address registers in the data bridge having a corresponding initial value and mask value that is stored in the read only memory.

6. The data bridge system according to claim 5, wherein the ASICs are graphic adapters, and wherein the initial values and the mask values stored in the read only memory define the base address registers in the data bridge as a function of the configuration requirements of the graphic adapters.

7. The data bridge system according to claim 1, wherein the data bridge forms base address registers as a function of the initial values and mask values stored in the read only memory, and wherein a first base address register defines prefetchable memory space, a second base register address defines non-prefetchable memory space, and a third base address register defines I/O mapped space.

8. The data bridge system according to claim 1, wherein the data bridge has a plurality of base address registers that are programmable as a function of the initial values and mask values in the read only memory.

9. The data bridge system according to claim 1, wherein the read only memory is at least one of removably coupled to the data bridge and writable.

10. A method for configuring a plurality of ASICs in a multi-ASIC system comprising:

reading, initial values and mask values from a read only memory; and

forming, from the initial values and the mask values, configurable registers, that upon initialization of a system cause the system to allocate resources to each of the plurality of ASICs.

11. The method according to claim 10,

wherein the computer system has an interface that, upon initialization of the computer system, writes all ones to base address registers of a data bridge and then reads back values to determine a size of a resource; and

wherein the computer system allocates memory space in the computer system as a function of the information in the base address registers.

12. The method according to claim 10, wherein the interface is connected to a north bridge in a computer system.

13. The method according to claim 10, wherein the data bridge has six base address registers, each of the base address registers in the data bridge having a corresponding initial value and mask value that is stored in the read only memory.

14. The method according to claim 13, wherein the ASICs are graphic adapters, and wherein the initial values and the mask values stored in the read only memory define the configuration registers in the data bridge as a function of the configuration requirements of the graphic processors.

15. The method according to claim 14, wherein the data bridge forms configuration registers as a function of the initial values and mask values stored in the read only memory, and wherein a first base address register defines prefetchable memory space, a second base register

address defines non-prefetchable memory space, and a third base address register defines I/O mapped space.

16. The method according to claim 10, wherein the data bridge has a plurality of base address registers that are programmable as a function of the initial values and mask values in the read only memory.

17. The method according to claim 10, wherein the read only memory is at least one of removably coupled to the data bridge and is writable.

18. The method according to claim 10, wherein at least one of the plurality of ASICs is a graphic adapter.

19. A data bridge system, comprising:

an interface for transferring data;

a plurality of components;

a data bridge operatively coupled to each of the interface and the plurality of components; and

the data bridge having a storage device for storing at least initial values and mask values for each component of the plurality of components.

20. The data bridge system according to claim 19, wherein at least one of the plurality of components is a graphic adapter.

21. The data bridge system according to claim 19, wherein the interface is a north bridge in a computer system.

22. The data bridge system according to claim 19, wherein the data bridge upon initialization forms basic address registers that are queried by the interface.

23. The data bridge system according to claim 22, wherein the data bridge has six base address registers, each of the base address registers in the data bridge having a corresponding initial value and mask value that is stored in the storage device.

24. The data bridge system according to claim 23, wherein the components are graphic adapters, and wherein the initial values and the mask values stored in the storage device define registers in the data bridge as a function of the configuration requirements of the graphic adapters.

25. The data bridge system according to claim 19, wherein the data bridge forms base address registers as a function of the initial values and mask values stored in the storage device, and wherein a first base address register defines prefetchable memory space, a second base register address defines non-prefetchable memory space, and a third base address register defines I/O mapped space.

26. The data bridge system according to claim 19, wherein the data bridge has a plurality of base address registers that are programmable as a function of the initial values and mask values in the storage device.

27. The data bridge system according to claim 19, wherein the read only memory is removably coupled to the data bridge.

28. A circuit comprising:

memory containing initial values and mask values for use in forming a register; and

at least one configurable register that includes register configuration logic and at least one register flop to contain an initial value and at least one mask flop that generates a mask bit for the configuration logic and wherein the register configuration logic configures the at least one register flop to be read and/or writable based on at least one mask value stored in the memory.

33. The method according to claim 10 wherein at least a plurality of the ASICs are graphic processors, and wherein the initial values and mask values stored in the read only memory define the configuration registers in the data bridge as a function of the configuration requirements of the plurality of graphic processors.

34. The data bridge system of claim 1 comprising at least one configurable register that includes register configuration logic and at least one register flop to contain an initial value and at least one mask flop that generates a mask bit for the configuration logic and wherein the

registered register configuration logic configures the at least one register flop to be read and/or writable based on at least one mask value stored in the memory.

35. The method according to claim 10 wherein forming the configurable registers comprises forming at least one register flop of the configurable register to be read and/or writable based on at least one mask value.

EVIDENCE APPENDIX

[NONE]

APPENDIX B

RELATED PROCEEDINGS

[NONE]

APPENDIX C